Sheaf codes: an expository introduction via the toric code

Cassie Ding

May 2026

Abstract

This note is an expository introduction to sheaf-theoretic coding theory, with the toric code as a running example. After fixing the basic vocabulary of linear codes — rate, distance, locality — we describe the toric code in pictorial language and reinterpret every piece of it as local linear data on the cellular chain complex of the torus. This rereading motivates the definitions of a *cellular sheaf* and a *sheaf code*, of which the toric code is the simplest example. We then explain why the formalism is forced on us as soon as one seeks quantum LDPC codes with constant rate and linear distance: the Bravyi–Poulin–Terhal bound rules out fixed-dimensional Euclidean constructions, and the modern qLDPC breakthroughs combine an expanding cell complex with nontrivial local-algebra stalks linked by restriction maps — precisely the structure of a cellular sheaf. A short tour reads each major construction (classical Tanner, DELLM, Leverrier–Zémor quantum Tanner, Dinur–Lin–Vidick higher cubical) as a cellular sheaf code on this common template.

1 What is a code?

1.1 Linear codes and parity checks

Fix a finite field $\mathbb{F} = \mathbb{F}_q$; readers happy with bits should imagine $\mathbb{F} = \mathbb{F}_2 = \{0, 1\}$. A *linear code* of length n is just a linear subspace

$$C \subseteq \mathbb{F}^n.$$

Vectors in C are called *codewords*. The dimension $k := \dim_{\mathbb{F}} C$ counts how many independent messages we can encode, and the *rate*

$$R(C) := \frac{k}{n}$$

measures how much of the channel is dedicated to information versus redundancy.

A code is usually specified by a *parity-check matrix* $H \in \mathbb{F}^{m \times n}$, in which case

$$C = \ker H = \{\, x \in \mathbb{F}^n : Hx = 0 \,\}.$$

Each row of H is one "local check" that a candidate vector must pass.

Example 1.1 (3-bit repetition code)**.** $n = 3$, $H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$, $C = \{000, 111\}$, $k = 1$, $R = 1/3$. *Each row of H says "two adjacent bits must agree".*

1.2 Distance and locality

Two more numbers control how usable a code is.

The *Hamming weight* $\mathrm{wt}(x)$ is the number of nonzero coordinates of x, and the *minimum distance* of C is

$$d(C) := \min\{\, \mathrm{wt}(x) : x \in C,\ x \neq 0 \,\}.$$

A code with minimum distance d can detect any error of weight $< d$ and correct any error of weight $< d/2$.[1]

The *locality* of C is how sparse the parity-check matrix H can be. We say C is *LDPC* (low-density parity-check) if there is a choice of H in which every row and every column has only $O(1)$ nonzero entries: each check inspects a constant number of bits, and each bit participates in only a constant number of checks.

Remark 1.2 (The three knobs)**.** *Rate R, relative distance $\delta := d/n$, and locality are the three fundamental knobs. "Good codes" have $R, \delta = \Omega(1)$ and constant locality; $\delta = \Omega(1)$ means the minimum distance d grows* linearly *in the block length n, so the fraction of errors the code can correct remains bounded below by a positive constant independent of n. Achieving all three at once classically is the Sipser–Spielman expander-code story [7, 8]. In the quantum world the same question is much harder, and is what eventually forces us to introduce sheaves.*

2 The toric code, pictorially

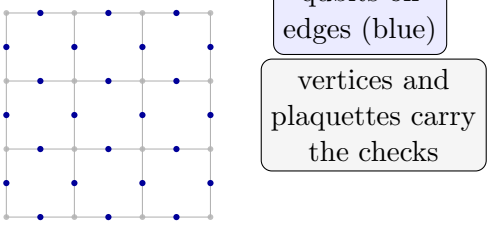
We now describe the central example that will run through the rest of these notes. It is the simplest interesting *quantum* code [2], and also the canonical introductory example of a quantum *sheaf* code: the one in which every stalk is $\mathbb{F}_2$ and every restriction is the identity, so that all the structure comes from the topology of the underlying complex.

2.1 The setup

Take an $L \times L$ square lattice on a torus (identify opposite sides of a big square). Put one qubit on every *edge*. The torus has

$$|V| = L^2 \text{ vertices}, \quad |E| = 2L^2 \text{ edges}, \quad |F| = L^2 \text{ plaquettes (faces)}.$$

So we have $n = |E| = 2L^2$ physical qubits.



[1] More precisely, a code of minimum distance d corrects all error patterns of Hamming weight t if and only if $2t+1 \le d$, i.e. $t \le \lfloor (d-1)/2 \rfloor$, which for integer t is equivalent to $t < d/2$. See e.g. [1] for a proof via the Hamming sphere-packing argument.

2.2 Two kinds of checks

A *plaquette* of the square lattice is a 2-cell, i.e. a unit square p of the lattice; its boundary ∂p is the set of four edges bounding it. "Plaquette" is the standard physics name for this 2-cell.

For every vertex v, the *star operator*

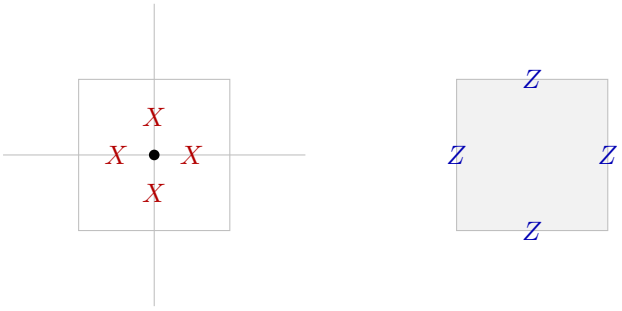
$$A_v := \prod_{e : v \in \partial e} X_e$$

applies the single-qubit bit-flip X to each of the four edges e incident to v.

For every plaquette p, the *plaquette operator*

$$B_p := \prod_{e \in \partial p} Z_e$$

applies the single-qubit phase-flip Z to each of the four edges in the boundary of p.



star A_v plaquette B_p

Why these commute. A_v and B_p share either 0 or 2 qubits, because a vertex and a plaquette either don't touch or share exactly two edges. Two X's and two Z's anticommute pairwise, and an even number of anticommutations is a commutation. So all A_v's commute with all B_p's, and we may impose them simultaneously.

Example 2.1 (Explicit commutation)**.** *On the $L = 2$ torus, label vertices by $(i, j) \in (\mathbb{Z}/2\mathbb{Z})^2$. Take $v = (0, 0)$ and $p =$ the unit square with corners $(0, 0), (1, 0), (1, 1), (0, 1)$.*

- *A_v acts with X on the four edges incident to $(0, 0)$: the rightward edge $e_1 = (0, 0) \to (1, 0)$, the upward edge $e_2 = (0, 0) \to (0, 1)$, and the two edges arriving from $(1, 0)$ and $(0, 1)$ via the torus identification.*
- *B_p acts with Z on the four boundary edges of p: e_1 (bottom), e_2 (left), $(1, 0) \to (1, 1)$ (right), $(0, 1) \to (1, 1)$ (top).*

The shared edges are e_1 and e_2 — exactly 2. On each shared edge X and Z anticommute, contributing a factor (-1) each; on the other $2 + 2$ edges the operators act on disjoint qubits and commute trivially. Total commutation factor: $(-1)^2 = +1$. Hence $A_v B_p = B_p A_v$. □

The code space. The *toric code* is the joint $+1$-eigenspace of all A_v and all B_p. On the $L \times L$ torus this space has dimension 4, i.e. it encodes $k_{\log} = 2$ logical qubits.[2]

[2] A *logical qubit* is an encoded qubit: the code space, here of complex dimension $2^{k_{\log}} = 4$, is isomorphic as a Hilbert space to $(\mathbb{C}^2)^{\otimes k_{\log}}$. Operations on logical qubits (logical gates) are physical operations that preserve the

Logical operators are loops. Logical operators split by Pauli type. A Z-type Pauli operator Z_S supported on a set S of edges commutes with a star A_v if and only if $|S \cap \mathrm{star}(v)|$ is even; it commutes with *every* star if and only if every vertex is incident to an even number of edges in S, i.e. S is a 1-*cycle* (an element of $\ker \partial_1$). Such a Z_S is not a product of plaquette operators B_p if and only if S is not a boundary (not in $\mathrm{im}\, \partial_2$), i.e. $[S]$ is a nonzero class in $H_1(T^2; \mathbb{F}_2)$. The minimum-weight representative of a nonzero class is a noncontractible loop.

Dually, an X-type Pauli operator X_T supported on a set T of edges commutes with every plaquette B_p if and only if T is a 1-*cocycle* (an element of $\ker \delta^1$), and is not a product of star operators if and only if $[T]$ is nonzero in $H^1(T^2; \mathbb{F}_2)$, i.e. T represents a noncontractible loop in the dual lattice.

The torus has two independent generators in each of $H_1(T^2; \mathbb{F}_2)$ and $H^1(T^2; \mathbb{F}_2)$, giving two independent logical operators $\overline{Z}_1, \overline{Z}_2$ and two independent $\overline{X}_1, \overline{X}_2$. Contractible 1-cycles are boundaries of plaquette regions and hence products of B_p; contractible 1-cocycles are coboundaries $\delta^0(f)$ for some $f \in C^0$ and hence products of A_v.

Rate, distance, locality. Putting the pieces together:

$$n = 2L^2, \qquad k_{\log} = 2, \qquad d = L, \qquad R = \frac{1}{L^2} \to 0, \quad \delta = \frac{1}{2L} \to 0.$$

Every check has weight 4 and every qubit participates in exactly 4 checks, so the code is LDPC. But neither the rate nor the relative distance is bounded below by a constant: the toric code has *vanishing* rate and relative distance. This is one of the central limitations that motivates the sheaf-code generalization.

3 Reading the toric code as "local linear data"

We now reread Section 2 in a way that has every ingredient labelled by a *cell* (a vertex, edge, or plaquette of the lattice). This is the bridge to sheaves.

3.1 Cellular chain complex of the torus

Let X be the cellular structure of the torus: cells $X(0), X(1), X(2)$ are vertices, edges, and plaquettes. Form the $\mathbb{F}_2$-vector spaces

$$C_0 = \mathbb{F}_2^{X(0)}, \quad C_1 = \mathbb{F}_2^{X(1)}, \quad C_2 = \mathbb{F}_2^{X(2)}.$$

The boundary maps

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0, \qquad \partial_2(p) = \sum_{e \in \partial p} e, \quad \partial_1(e) = \sum_{v \in \partial e} v,$$

satisfy $\partial_1 \partial_2 = 0$ (boundary-of-boundary). Dually we have the coboundary

$$C^0 \xrightarrow{\delta^0} C^1 \xrightarrow{\delta^1} C^2, \qquad \delta^0 = \partial_1^T, \quad \delta^1 = \partial_2^T, \qquad \delta^1 \delta^0 = 0.$$

code space while implementing nontrivial actions on its $2^{k_{\log}}$-dimensional structure. Information stored in a logical qubit is protected against any physical error whose support is too small to implement a nontrivial logical operator.

3.2 Stabilizers as boundary maps

- Reading $\delta^0 : C^0 \to C^1$ column by column, $\delta^0(v)$ is the indicator vector of the four edges meeting v — exactly the X-stabilizer at v, viewed as a vector on the edges. Stacking these as *rows* gives the CSS parity-check matrix

$$H_X = (\delta^0)^T = \partial_1.$$

- Similarly, $\partial_2(p) \in C_1$ is the indicator vector of the four edges around p — the Z-stabilizer at p. Stacking these as rows gives

$$H_Z = \partial_2^T = \delta^1.$$

The CSS commutation condition $H_X H_Z^T = 0$ is then $\partial_1 \partial_2 = 0$, i.e. "$\partial^2 = 0$"; transposing gives the dual identity $\delta^1 \delta^0 = 0$.

Remark 3.1 (Why CSS commutation is necessary)**.** *For a CSS code to define a valid stabilizer code, every X-type stabilizer must commute with every Z-type stabilizer. For Pauli operators, X_S and Z_T (supported on edge sets S and T respectively) commute if and only if $|S \cap T|$ is even. In matrix language, if $h_X \in \mathbb{F}_2^n$ is a row of H_X and $h_Z \in \mathbb{F}_2^n$ is a row of H_Z, then X_{h_X} and Z_{h_Z} commute iff $h_X \cdot h_Z = 0$ (inner product over $\mathbb{F}_2$). The condition $H_X H_Z^T = 0$ says this holds* for every pair of rows simultaneously*: the X-checks and Z-checks form a mutually commuting family, so they can all be imposed on the same Hilbert space. Without this condition the "stabilizer group" would contain anticommuting elements, which cannot be simultaneously diagonalized.*

3.3 Logical operators as (co)homology

The CSS quotient spaces

$$\ker H_Z / \operatorname{im} H_X^T \;\cong\; H^1(X; \mathbb{F}_2), \qquad \ker H_X / \operatorname{im} H_Z^T \;\cong\; H_1(X; \mathbb{F}_2)$$

are precisely the first cellular cohomology and homology of the torus. On the torus both are $\mathbb{F}_2^2$, recovering the $k_{\log} = 2$ logical qubits and matching the two independent noncontractible loops. The distance is the minimum support of a nonzero (co)homology representative, i.e. the systolic distance L.

Remark 3.2 (Slogan)**.** *The toric code is the CSS code [3, 4] of the cellular chain complex $C_\bullet(\text{torus}; \mathbb{F}_2)$. "Stabilizers = boundaries", "logicals = (co)homology", "commutation = $\partial^2 = 0$".*

4 Cellular sheaves

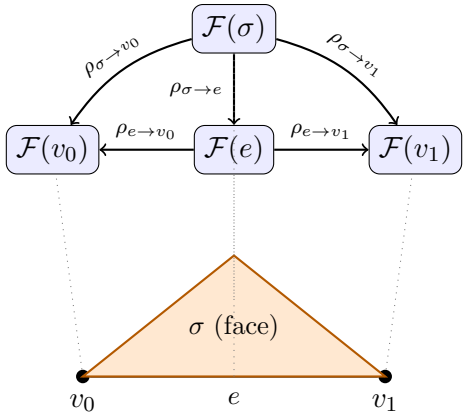
The toric code uses the constant coefficient field $\mathbb{F}_2$ at every cell. If we want to upgrade the construction — attach a richer vector space at each cell, encode tensor-code or Reed–Solomon constraints into the local data, or mix them across dimensions — we need a way to say "the coefficient space can vary cell to cell, with consistent restriction maps". This is the data of a *cellular sheaf* (in the coding-theory literature; algebraic topologists would describe it as a contravariant functor from the face poset P of X to $\mathbb{F}$-vector spaces, equivalently a covariant functor from P^{op}, since restriction maps go from larger cells to smaller faces and therefore reverse the face-poset order; a *cosheaf* would be the covariant functor from P itself, with maps going upward from faces to cofaces). See [9] for a general introduction to cellular sheaves and their spectral theory.

4.1 Definition by pictures

Definition 4.1 (Cellular sheaf, working definition)**.** *Let X be a finite graded cell complex with cells $X(0), X(1), \dots, X(t)$. A* cellular sheaf $\mathcal{F}$ *over a field $\mathbb{F}$ consists of:*

1. *a finite-dimensional $\mathbb{F}$-vector space $\mathcal{F}(\sigma)$ for every cell $\sigma \in X$;*
2. *for every strict face relation $\tau < \sigma$ (i.e. τ is a proper face of σ), a linear* restriction map
$$\rho_{\sigma \to \tau} : \mathcal{F}(\sigma) \to \mathcal{F}(\tau);$$
and for $\tau = \sigma$, the restriction is defined to be the identity: $\rho_{\sigma \to \sigma} := \mathrm{id}_{\mathcal{F}(\sigma)}$;
3. *compatibility: whenever $\tau' \le \tau \le \sigma$, $\rho_{\tau \to \tau'} \circ \rho_{\sigma \to \tau} = \rho_{\sigma \to \tau'}$.*

So a sheaf is just "one little vector space per cell, plus a rule for restricting from a face to any of its sub-faces, plus a transitivity condition".



Each cell carries its own vector space; restriction maps go from larger cells to their faces. Transitivity says that going $\sigma \to e \to v_0$ and $\sigma \to v_0$ directly give the same map.

The cellular cochain complex is

$$C^k(X; \mathcal{F}) = \bigoplus_{\sigma \in X(k)} \mathcal{F}(\sigma),$$

A remark on conventions: our restriction maps $\rho_{\sigma \to \tau}$ go from larger cells down to their faces, which is the standard sheaf direction (stalks restrict from higher-dimensional cells to lower-dimensional faces, exactly as in Hansen–Ghrist [9]). These maps define the *boundary* operator $\partial_{k+1} : C_{k+1}(X; \mathcal{F}) \to C_k(X; \mathcal{F})$ by

$$(\partial_{k+1}\, c)_\tau \;=\; \sum_{\substack{\sigma \in X(k+1) \\ \tau \le \sigma}} [\sigma : \tau]\; \rho_{\sigma \to \tau}(c_\sigma), \qquad \tau \in X(k),$$

where $[\sigma : \tau] \in \{\pm 1\}$ is the signed incidence number and $c_\sigma \in \mathcal{F}(\sigma)$; over $\mathbb{F}_2$ all signs are 1. The incidence numbers depend on a choice of orientation for each cell (a linear ordering of the vertices of each simplex, or an orientation of each cell in the CW sense); once orientations are fixed, $[\sigma : \tau]$ is $+1$ if the induced orientation on τ from σ agrees with the chosen orientation of τ, and -1 otherwise. Different choices of orientation change the signs of some rows and columns of ∂_{k+1} but leave $\ker \partial_{k+1}$ and $\mathrm{im}\, \partial_{k+1}$ (and hence all cohomology groups) invariant. Over $\mathbb{F}_2$ the sign choice is irrelevant and all $[\sigma : \tau] = 1$.

Example 4.2 (Standard orientation for the square lattice)**.** *On the $L \times L$ torus: orient each horizontal edge left-to-right and each vertical edge bottom-to-top (coordinates mod L); orient each plaquette counterclockwise. With $e^h_{i,j}$ the horizontal edge starting at (i,j) and $e^v_{i,j}$ the vertical edge starting at (i,j), the signed boundary of the plaquette p_{ij} with bottom-left corner (i,j) is*

$$\partial_2(p_{ij}) = e^h_{i,j} + e^v_{i+1,j} - e^h_{i,j+1} - e^v_{i,j} \qquad (\text{over } \mathbb{Z}).$$

Over $\mathbb{F}_2$ the signs vanish and this reduces to the indicator of the four boundary edges, matching the plaquette operator B_p exactly.

This is well-typed: $\rho_{\sigma\to\tau} : \mathcal{F}(\sigma) \to \mathcal{F}(\tau)$ is applied to $c_\sigma \in \mathcal{F}(\sigma)$, yielding an element of $\mathcal{F}(\tau)$. Choosing a basis on each stalk identifies $C^k(X;\mathcal{F})$ with $C_k(X;\mathcal{F})$, and the *coboundary* is then $\delta^k := \partial_{k+1}^T$, i.e. the matrix transpose of ∂_{k+1} under these bases. One verifies $\partial_k\partial_{k+1} = 0$ from two independent ingredients. Fix $\nu \in X(k-1)$. The ν-component of $\partial_k(\partial_{k+1}c)$ expands as

$$(\partial_k\partial_{k+1}c)_\nu = \sum_{\substack{\sigma\in X(k+1)\\ \tau:\,\nu\le\tau\le\sigma}} [\sigma:\tau][\tau:\nu]\,\rho_{\tau\to\nu}\big(\rho_{\sigma\to\tau}(c_\sigma)\big).$$

Ingredient (ii): sheaf compatibility gives $\rho_{\tau\to\nu}\circ\rho_{\sigma\to\tau} = \rho_{\sigma\to\nu}$, so the double sum collapses to

$$(\partial_k\partial_{k+1}c)_\nu = \sum_{\sigma\in X(k+1)} \Big(\sum_{\tau:\,\nu\le\tau\le\sigma} [\sigma:\tau][\tau:\nu]\Big)\rho_{\sigma\to\nu}(c_\sigma).$$

Ingredient (i): the incidence arithmetic of any oriented cell complex gives $\sum_{\tau:\,\nu\le\tau\le\sigma}[\sigma:\tau][\tau:\nu] = 0$ for every pair (σ,ν) with $\dim\sigma = \dim\nu + 2$, so every term in the outer sum vanishes.

Remark 4.3 (Recovery of Hansen–Ghrist formalism)**.** *Transposing $\partial_k\partial_{k+1} = 0$ gives $\delta^k\delta^{k-1} = 0$. The resulting cochain complex $\cdots \to C^{k-1}(X;\mathcal{F}) \xrightarrow{\delta^{k-1}} C^k(X;\mathcal{F}) \xrightarrow{\delta^k} C^{k+1}(X;\mathcal{F}) \to \cdots$ is precisely the sheaf-cohomology complex of Hansen–Ghrist [9]. The cohomology groups $H^k(X;\mathcal{F}) := \ker\delta^k/\operatorname{im}\delta^{k-1}$ are the fundamental invariants of the sheaf; for a constant sheaf they recover the ordinary cellular cohomology groups $H^k(X;\mathbb{F})$.*

4.2 The constant sheaf and the toric code

Example 4.4 (Constant sheaf)**.** *The* constant sheaf $\underline{\mathbb{F}}$ *assigns $\mathbb{F}$ to every cell and the identity to every restriction. Its cochain complex is the ordinary cellular cochain complex of X. On the torus with $\mathbb{F} = \mathbb{F}_2$, this is exactly the chain complex underlying the toric code.*

So in this language, the toric code is the CSS code of the *constant* sheaf $\underline{\mathbb{F}_2}$ on the torus.

4.3 Intuitive picture: constant vs. nontrivial

It is worth fixing a single mental picture of the constant-vs-nontrivial distinction before turning to formal sheaf codes.

Constant sheaf codes. Every cell carries the *same* vector space V, and every restriction map is the identity on V. The cell complex supplies the *geometry* (which cells meet which); the *algebra* is a single fixed coefficient space V replicated at every cell. The toric code uses $V = \mathbb{F}_2$ on the torus: one bit per cell, identity restrictions. The only thing that varies from cell to cell is the *address* of the cell within the complex, not the local algebra.

Nontrivial sheaf codes. Different cells carry *different* vector spaces, and the restrictions are nontrivial linear maps relating them. The geometry still comes from the cell complex, but the local algebra now varies cell by cell. Two conventions appear in the literature and both are valid; the paper uses both and the reader should distinguish them.

Convention A (stalk = local code). Each stalk $\mathcal{F}(\sigma)$ is itself a local code space — e.g. the degree-r polynomial space on a cell-specific $\mathbb{P}^1$ (whose image under evaluation is a Reed–Solomon code), a BCH code, or a tensor-product code — and the restriction maps are the evaluation or projection maps that the code family supplies. The global code $\ker \delta^\ell$ then arises from the cocycle condition on these local code stalks.

Convention B (stalk = ambient space; code = local check). Each stalk $\mathcal{F}(\sigma)$ is an ambient vector space (e.g. $\mathbb{F}_q^\Delta$ at an edge, $\mathbb{F}_q^\Delta \otimes \mathbb{F}_q^\Delta$ at a face), and the local code $C \subseteq \mathcal{F}(\sigma)$ is encoded into the sheaf via the restriction maps: the coboundary $(\delta^\ell c)_\sigma$ is designed to vanish precisely when the restrictions of c to the faces of σ assemble into a codeword of C. The global code is therefore $\ker \delta^\ell$ exactly as in Convention A — the local-code checks are *not* imposed as a separate layer on top of $\ker \delta^\ell$ but are the very conditions that define it.

The left-right Cayley complex constructions of §7 use Convention B (see the U2 description): the face stalk is the full bilinear space $\mathbb{F}_q^{\Delta^2}$, and $C_A \otimes C_B$ is a check constraint, not the stalk. The single-square worked example of §8 uses Convention A: the stalks are polynomial evaluation spaces. Whichever convention is adopted, the sheaf records how to compare the local data at adjacent cells via the restriction maps.

Two refinements of the naive picture. A natural first guess at this distinction is "every element of the cochain complex carries its own coefficient system." The unit of variation is actually the *cell*, not the individual cochain entry. A cochain $x \in C^k(X;\mathcal{F})$ assigns one vector $x_\sigma \in \mathcal{F}(\sigma)$ independently to each k-cell σ, with *no* compatibility condition; the cochain space therefore splits as the direct sum

$$C^k(X;\mathcal{F}) \;=\; \bigoplus_{\sigma \in X(k)} \mathcal{F}(\sigma).$$

(An element of $H^0(X;\mathcal{F})$, the *global sections*, is the restricted class of cochains satisfying full compatibility; a general cochain imposes no such constraint.) The stalk attached to cell σ is the local code at σ; the components of a cochain are independently chosen entries of those local codes.

Global sections for a code. Since global sections will play a specific role in coding applications, we state the condition explicitly. An element $x \in C^0(X;\mathcal{F}) = \bigoplus_{v \in X(0)} \mathcal{F}(v)$ is a *global section* ($x \in H^0(X;\mathcal{F}) = \ker \delta^0$) if and only if for every edge e with endpoints v_0, v_1:

$$\rho^T_{e \to v_1}(x_{v_1}) - \rho^T_{e \to v_0}(x_{v_0}) = 0 \;\in\; \mathcal{F}(e),$$

i.e. the stalk data at adjacent vertices *agree* in the sense prescribed by the restriction maps. For the constant sheaf this forces x to be constant on connected components (recovering $H^0(X;\mathbb{F}) \cong \mathbb{F}^{\pi_0(X)}$); for a non-constant sheaf it imposes richer restriction-compatible constraints. A general cochain assigns values to vertices freely, with no such compatibility; the global sections are the special cochains that "fit together" across every edge.

Second, although nothing in the formal definition forbids mixing wildly different code families (RS at one cell, BCH at the next), in practice the stalks of one sheaf are drawn from a single *parameterized* family — "degree-r polynomials on $\mathbb{P}^1_\sigma$" for varying r and varying σ, "evaluations

of a fixed code on the labels visible from σ," and so on. Coherence inside a parameterized family is what makes the restriction maps natural rather than ad-hoc: each restriction is the map that the family supplies, not a separately invented linear map.

Slogan. A constant sheaf code is "one local code, replicated everywhere"; a nontrivial sheaf code is "a parameterized family of local codes, one per cell, glued by the family's own restriction maps." The toric code lives at the constant end; modern qLDPC and transversal-gate constructions live at the nontrivial end, with RS/BCH-style stalks varying cell by cell.

5 Sheaf codes

5.1 Classical sheaf codes

Definition 5.1 (Sheaf code)**.** *Let X be a finite cell complex with a cellular sheaf $\mathcal{F}$ over $\mathbb{F}$. Fix a* degree $\ell \geq 0$. *The associated* sheaf code *is*

$$C^\ell(X;\mathcal{F}) \xrightarrow{\delta^\ell} C^{\ell+1}(X;\mathcal{F}), \qquad \mathrm{Code}_\ell(X,\mathcal{F}) := \ker \delta^\ell.$$

Words are cellular ℓ-cocycles; the parity-check matrix is δ^ℓ.

Remark 5.2 (Notation)**.** *We use ℓ for the cochain degree throughout §5 to avoid collision with $k := \dim_{\mathbb{F}} \ker \delta^\ell$, the classical code dimension, and with $k_{\log} := \dim H^\ell(X;\mathcal{F})$, the number of logical qubits in the quantum setting.*

Rate, distance, and locality translate directly:

Definition 5.3 (Code parameters of a sheaf code)**.** *Let $\mathrm{Code}_\ell(X,\mathcal{F}) = \ker \delta^\ell$ be a sheaf code of degree ℓ.*

(N) Block length. $n = \sum_{\sigma \in X(\ell)} \dim \mathcal{F}(\sigma)$.

(R) Rate. $R = \dim \ker \delta^\ell / n$.

(D) Distance. *Two distinct notions are in common use and should not be conflated. The* cocycle distance *is $d_{\mathrm{cyc}} := \min\{\mathrm{wt}(x) : x \in \ker \delta^\ell,\ x \neq 0\}$. The* cohomological distance *is $d_{\mathrm{coh}} := \min\{\mathrm{wt}(x) : x \in \ker \delta^\ell \backslash \mathrm{im}\, \delta^{\ell-1},\ x \neq 0\}$, the minimum weight of a representative of a nonzero cohomology class. For a classical sheaf code the relevant distance is d_{cyc}; for the associated quantum CSS code (§5.2) the relevant distance is d_{coh}. For the toric code: $d_{\mathrm{cyc}} = 4$ (weight of a vertex coboundary $\delta^0(e_v) \in \ker \delta^1$) and $d_{\mathrm{coh}} = L$ (systolic length of $H^1(T^2;\mathbb{F}_2)$); it is $d_{\mathrm{coh}} = L$ that is quoted as the toric-code distance throughout. We use the* stalkwise *weight convention $\mathrm{wt}(x) = \sum_\sigma \mathrm{wt}(x_\sigma)$; an alternative* cellwise *convention $\mathrm{wt}(x) = \#\{\sigma : x_\sigma \neq 0\}$ differs by at most a stalk-dimension factor. Beyond the systole, $H^\ell(X;\mathcal{F})$ may carry torsion and p-adic precision data studied in [10].*

(L) Locality. *Each row of δ^ℓ inspects only the ℓ-faces of one $(\ell{+}1)$-cell. If X has bounded local degree and stalks have bounded dimension, the code is LDPC.*

5.2 Quantum (CSS) sheaf codes

A CSS code [3, 4] wants *two* parity-check matrices satisfying $H_X H_Z^T = 0$. Sheaves supply this for free: a chain complex.

Definition 5.4 (Quantum sheaf code)**.** *Suppose the cellular sheaf $\mathcal{F}$ on X comes with a cochain complex*

$$\cdots \to C^{\ell-1}(X;\mathcal{F}) \xrightarrow{\delta^{\ell-1}} C^{\ell}(X;\mathcal{F}) \xrightarrow{\delta^{\ell}} C^{\ell+1}(X;\mathcal{F}) \to \cdots, \qquad \delta^{\ell}\delta^{\ell-1} = 0.$$

The associated CSS sheaf code *places qubits on C^ℓ and uses*

$$H_X = (\delta^{\ell-1})^T, \qquad H_Z = \delta^{\ell},$$

so that the CSS commutation $H_X H_Z^T = 0$ is exactly $\delta^\ell\delta^{\ell-1} = 0$. The logical space is the cohomology $H^\ell(X;\mathcal{F})$; the number of logical qubits is $k_{\log} := \dim H^\ell$, and the X- and Z-distances are the minimum supports of nonzero classes in H^ℓ and its chain-complex dual, respectively.

5.3 The toric code as a CSS sheaf code

Let us walk the definition through the toric-code example one last time. The cell complex is

$$X = \text{torus cell structure:} \quad X(0) = \text{vertices}, \quad X(1) = \text{edges}, \quad X(2) = \text{plaquettes},$$

with $|X(0)| = L^2$, $|X(1)| = 2L^2$, $|X(2)| = L^2$ on the $L \times L$ torus. The sheaf is the constant sheaf $\mathcal{F} = \underline{\mathbb{F}_2}$ at degree $\ell = 1$:

$$\mathcal{F}(\sigma) = \mathbb{F}_2 \text{ for all } \sigma, \quad \rho_{\sigma\to\tau} = \mathrm{id} \text{ for all } \tau \le \sigma.$$

Then

$$C^0 = \mathbb{F}_2^V,\ C^1 = \mathbb{F}_2^E,\ C^2 = \mathbb{F}_2^F,$$

and

$$H_X = (\delta^0)^T = \partial_1, \qquad H_Z = \delta^1 = \partial_2^T.$$

Each row of H_X is one vertex star; each row of H_Z is one plaquette; $H_X H_Z^T = \partial_1\partial_2 = 0$ since $\partial^2 = 0$. The logical qubits live in $H^1(X;\mathbb{F}_2) = \mathbb{F}_2^2$, giving $k_{\log} = 2$; their representatives are noncontractible loops. Distance $d_{\mathrm{coh}} = L$, rate $k_{\log}/n = 1/L^2$, locality 4.

What was special about the toric code. All stalks $\mathcal{F}(\sigma) = \mathbb{F}_2$ were one-dimensional. All restriction maps were the identity on $\mathbb{F}_2$. All the global structure was carried by the cell complex (the torus). This is why one can build the toric code without ever mentioning sheaves — but it is also why the toric code's parameters $(R, \delta) = (1/L^2,\ 1/(2L))$ are so far from constant.

6 Why do we need sheaves?

The toric code has constant locality, but its rate and distance both vanish. Two natural questions arise:

1. Can we keep constant locality and obtain *constant* rate?
2. Can we keep constant locality and obtain *linear* distance?

6.1 The Euclidean distance ceiling

The first reason has a name: the *Bravyi–Poulin–Terhal (BPT) bound* [6]. Suppose a stabilizer code has n qubits arranged on a D-dimensional Euclidean lattice, with every parity check supported inside a ball of radius $O(1)$. Then there is an absolute constant c_D with

$$k_{\log}\, d^{\,2/(D-1)} \;\le\; c_D\, n.$$

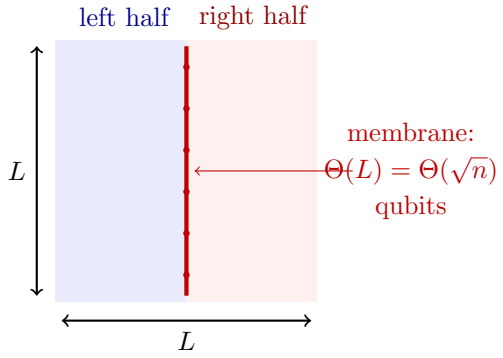
In two dimensions this reads $k_{\log}\, d^2 = O(n)$; in three dimensions, $k_{\log}\, d = O(n)$. A complementary $k_{\log}$-free distance ceiling in the same setting,

$$d \;\le\; O\big(n^{(D-1)/D}\big),$$

was established earlier by Bravyi–Terhal [5]. For *topological* stabilizer codes on a fixed D-manifold, $k_{\log}$ is additionally bounded by a Betti number of the manifold — a topological fact, independent of the BPT inequality.

Where the ceiling comes from. The two bounds rest on related but distinct arguments. For the Bravyi–Terhal distance bound $d \le O(n^{(D-1)/D})$: a logical operator must be supported on a region that no finite collection of local stabilizer blobs can patch over — intuitively, a region that *cuts through* the entire block. On a D-dimensional lattice the smallest such region is a $(D-1)$-dimensional *membrane* cleaving the block in half. A length-L cube has minimum cross-section $L^{D-1} = n^{(D-1)/D}$ qubits, and every logical operator must have weight at least this large. On the $L \times L$ torus the minimum membrane is a noncontractible loop of length $\Theta(L) = \Theta(\sqrt{n})$, which is exactly the toric-code distance $d = L$.

The joint bound $k_{\log} d^{2/(D-1)} \le c_D n$ of BPT requires additional work beyond the membrane argument. Roughly: there are $k_{\log}$ independent logical operators that must be simultaneously supported on disjoint (or nearly disjoint) membranes, and an isoperimetric argument on the stabilizer group — counting how much of the lattice must be "used up" to accommodate $k_{\log}$ independent membranes of weight $\ge d$ — yields the joint inequality. The membrane picture supplies the correct intuition for the distance bound; the BPT product bound on $k_{\log} d^{2/(D-1)}$ is the technically harder statement.



A logical operator on a 2D block must cleave the block in half. The smallest such cut is a 1D membrane of length $\Theta(\sqrt{n})$, which upper-bounds the distance.

The toric code saturates BPT. For the toric code, $D = 2$, $k_{\log} = 2$, $n = 2L^2$, $d = L$, so

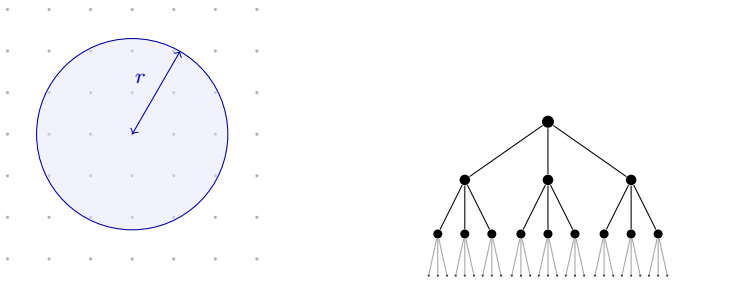
$$k_{\log}\, d^2 \;=\; 2L^2 \;=\; n,$$

which sits exactly on the BPT ceiling. Two consequences:

1. Stacking m independent toric codes on $n' = 2mL^2$ qubits yields $k'_{\log} = 2m$ and $d' = L$, so the rate $k'_{\log}/n' = 1/L^2$ remains vanishing regardless of m, and the relative distance $\delta = L/(2mL^2)$ shrinks as m grows. The BPT product $k'_{\log} d'^2 = 2mL^2 = n'$ stays on the BPT ceiling, confirming that this stacking strategy cannot simultaneously achieve constant rate and linear distance.

2. Increasing the dimension partially relaxes the distance ceiling (in $D = 3$ the Bravyi–Terhal bound gives $d \le O(n^{2/3})$ rather than $d \le O(\sqrt{n})$), but the rate stays $k_{\log}/n \to 0$ because $k_{\log}$ is bounded above by a fixed Betti number of the manifold — a topological invariant independent of the lattice refinement.

Why expanders escape it. The proof of BPT crucially uses that balls of radius r in a Euclidean lattice contain $\Theta(r^D)$ qubits — "polynomial growth." A bounded-degree *expander* graph (and the left-right Cayley complex built on one) has a different growth regime: for a Δ-regular graph of girth g, the absence of short cycles forces strict no-backtracking, so balls of radius $r < g/2$ grow as $\Theta((\Delta-1)^r)$ — matching the ball growth of the Δ-regular tree exactly up to the girth radius. For Ramanujan graphs and similar optimal constructions the girth satisfies $g = \Omega(\log_\Delta n)$, so this exponential growth persists for $r = \Omega(\log n)$. A related but independent structural fact is that families of bounded-degree expanders admit no bilipschitz embedding into any fixed-dimensional Euclidean space; this follows from the nonvanishing spectral gap together with Poincaré-type inequalities in ℓ_2, which show that high-girth expander families cannot be uniformly approximated by subsets of $\mathbb{R}^D$ for any fixed D. Geometrically local checks on such a complex are therefore not constrained by the small-membrane argument: codewords can have weight linear in n while every check still inspects only $O(1)$ qubits.



Euclidean lattice: $|B_r| = \Theta(r^D)$ expander / tree: $|B_r| = \Theta(\Delta^r)$

Polynomial vs. exponential ball growth. The BPT membrane argument uses polynomial growth; expander complexes have no such small-membrane obstruction, so linear-distance qLDPC codes become possible.

This is the deep reason every modern good qLDPC construction — Panteleev–Kalachev [11], Leverrier–Zémor quantum Tanner [12], the DELLM locally testable codes [13] — lives on an expander-type complex rather than on a fixed-dimensional manifold. The role of sheaves is then to attach genuinely interesting local algebra (Reed–Solomon, tensor-product codes) to the cells of that non-Euclidean complex.

Concretely, modern good qLDPC constructions need at least three independent upgrades over the toric code, all of which are most cleanly stated in sheaf language:

(U1) *Expanding cell complex.* Replace the torus by a bounded-degree complex (e.g. left-right Cayley square complex) whose links and global walks have spectral gaps. This kills the Euclidean-geometry distance ceiling.

(U2) *Tensor-product local codes.* Introduce nontrivial local algebra at each cell, replacing the single bit of the constant sheaf with a richer code structure. In the left-right Cayley square complex, the stalk at each *vertex* is $\mathbb{F}_q$ (one symbol per group element, the lowest-level coefficient), the stalk at each incident *edge* is the ambient symbol space $\mathbb{F}_q^{\Delta}$ (dimension Δ), and the stalk at each *face* (square) is the ambient bilinear space $\mathbb{F}_q^{\Delta} \otimes \mathbb{F}_q^{\Delta}$ (dimension Δ^2); the tensor-product code $C_A \otimes C_B \subseteq \mathbb{F}_q^{\Delta^2}$ is then imposed as a *local check* at each face, not as the stalk itself. Restrictions are given by row and column projections from the face ambient space to the edge ambient spaces, and by evaluation maps from edge stalks to the vertex stalk. This is exactly what cellular sheaves with non-constant stalks are for.

(U3) *Restriction-compatible structure on the chain complex.* The two-sided distance bounds and local testability are proven by inducting up dimensions, where each step relies on the restriction maps in the sheaf having "good local algebra" (surjective restrictions, bounded-support fillings). Constant sheaves cannot encode this — but cellular sheaves can.

A useful one-line summary of "why sheaves?" is therefore:

> *Sheaves provide a common framework in which the local checks themselves carry nontrivial algebraic content while remaining glued together by the global combinatorics of a cell complex — a combination that neither constant-coefficient codes nor standalone local codes can achieve.*

The toric code is the boundary case where "nontrivial algebraic content" collapses to a single bit per cell; everything beyond it lives genuinely in the sheaf world.

7 Reading the qLDPC breakthroughs as sheaf codes

The summary slogan of the previous section was that sheaves are needed because constant-coefficient codes on fixed-dimensional manifolds run into the BPT ceiling, while modern qLDPC constructions rely on local checks that are themselves nontrivial tensor-product or Reed–Solomon codes glued by a cellular sheaf. We now make that statement concrete by walking through each major construction and exhibiting it as a cellular sheaf code. The bottom line:

> *Every modern good qLDPC construction is the same composite — (non-Euclidean expander-type cell complex) × (nontrivial-algebra local stalks) — glued by the restriction maps of a cellular sheaf. Drop either factor and the construction collapses.*

7.1 Two failure modes for constant coefficients

Constant-coefficient codes fail to give good qLDPC parameters in two genuinely different ways.

Failure mode A: Euclidean geometry, constant stalks. This is the toric code (§ 2) and its higher-dimensional cousins. The cell complex is a fixed-dimensional manifold; the stalks are all $\mathbb{F}_2$. The BPT ceiling of § 6.1 then forces $k_{\log} d^{2/(D-1)} = O(n)$, and for topological codes specifically $k_{\log}$ is moreover bounded by a Betti number of the manifold. Neither constant rate nor linear distance is reachable.

Failure mode B: Expander geometry, constant stalks. What if we keep the constant sheaf but replace the manifold by an expander complex? On a graph (a 1-dimensional cell complex) the constant sheaf $\underline{\mathbb{F}_2}$ yields the cochain complex $C^0 = \mathbb{F}_2^V \xrightarrow{\delta^0} C^1 = \mathbb{F}_2^E$, with no further coboundary (there are no 2-cells). A CSS quantum code built from this complex places qubits on edges, uses $H_X = (\delta^0)^T$ (vertex-star checks) and $H_Z = 0$ (no Z-checks), giving a code that detects nothing with Z errors — not an interesting quantum code. Alternatively, reading $\ker \delta^0$ as a classical code on vertices gives the global sections: functions constant on connected components, a $[|V|, 1, |V|]$ repetition code for a connected graph. This code has *maximal* distance $|V|$ but *vanishing* rate $1/|V| \to 0$, and no amount of spectral expansion changes either figure: both the rate $1/|V|$ and the distance $|V|$ are determined purely by topology (the zeroth Betti number $\beta_0 = \dim H^0(G; \mathbb{F}_2)$, the number of connected components), not by the expansion constant. The fundamental problem is that a single bit per edge or vertex carries no algebraic structure for the expander geometry to act on.

For the *edge-variable* classical code on the expander graph (bits on edges, one parity check $[\Delta, \Delta - 1, 2]$ at each vertex), the minimum weight of a nonzero codeword equals the girth g of the underlying graph. By the Moore bound, any Δ-regular graph on $|V|$ vertices satisfies $g \le 2 \log_{\Delta-1} |V| + O(1)$, so the relative distance is $g/|E| \le g/(\Delta|V|/2) = O(\log n/n) \to 0$. This *girth bound* is tight for Ramanujan graphs but cannot be improved past $O(\log n/n)$ for any fixed-degree family; no expander property can push the relative distance above $O(\log n/n)$ when the local constraint is a single parity check.

For a *vertex-variable* Tanner code (bits on left nodes, local code C_0 of minimum distance d_0 at each right check), the Sipser–Spielman theorem [8] gives a quantitative lower bound. Let $G = (L \cup R, E)$ be a bipartite graph in which every left node has degree d, every right node has degree r, and every subset $S \subseteq L$ with $|S| \le n/2$ satisfies $|N(S)| \ge (1 - \epsilon) d\,|S|$ (where $n = |L|$). If $d_0 > 2\epsilon r$, the Tanner code has global minimum distance $\ge d_0\, n/(2r)$, giving

$$\delta_{\text{global}} \;\ge\; \frac{d_0}{2r} \;=:\; \frac{\delta_0}{2}, \qquad \delta_0 := \frac{d_0}{r}.$$

For a Δ-regular bipartite Tanner graph ($d = r = \Delta$) with a single parity check ($d_0 = 2$): $\delta_0 = 2/\Delta$, giving $\delta_{\text{global}} \ge 1/\Delta \to 0$ as $\Delta \to \infty$. Spectral expansion amplifies local distance only when $\delta_0 = \Omega(1)$, i.e. when the local code already has constant relative distance. The reason classical Tanner codes on expander graphs achieve linear distance at all is that the local code $C \subseteq \mathbb{F}_q^\Delta$ satisfies $d(C)/\Delta = \Omega(1)$, i.e. $d(C) \ge 3$ and more. Without this algebraic content the expander geometry gives nothing.

So we need both: a non-Euclidean cell complex *and* nontrivial local algebra at the stalks, coordinated by the restriction maps of a cellular sheaf.

7.2 A tour: modern constructions as sheaf codes

For each construction we record only the cell complex X, the stalks, and what makes the sheaf nontrivial. The cited papers are the authoritative sources for everything else.

- **Sipser–Spielman / classical Tanner codes** [7, 8]. X is a bounded-degree expander graph. Set edge stalk $\mathcal{F}(e) = \mathbb{F}_q$ and vertex stalk $\mathcal{F}(v) = \mathbb{F}_q^\Delta/C$, the quotient of the ambient symbol space by the local code $C \subseteq \mathbb{F}_q^\Delta$. The restriction $\rho_{e\to v} : \mathbb{F}_q \to \mathbb{F}_q^\Delta/C$ is the composition of the coordinate embedding $\mathbb{F}_q \hookrightarrow \mathbb{F}_q^\Delta$ (placing the edge symbol in the slot indexed by e) with the quotient projection. The sheaf boundary operator then gives the *unsigned* assembly map

$$(\partial_1^{\text{Tanner}} c)_v = \sum_{e \ni v} \rho_{e\to v}(c_e) \ \in \ \mathbb{F}_q^\Delta/C,$$

  where the sum carries no signed incidence numbers. (The standard Tanner code is orientation-independent: it depends only on which edges are incident to v, not on a chosen orientation. The signed cellular boundary ∂_1 of §4 coincides with this unsigned assembly over $\mathbb{F}_2$ — where $-1 = 1$ — but differs for $q > 2$, where using signed incidence numbers would make the "code" vary with the orientation of the graph.) The Tanner code is $\ker \partial_1^{\text{Tanner}} \subseteq C_1 = \mathbb{F}_q^E$: an edge assignment c is a codeword if and only if the assembled vector at every vertex is zero in the quotient, i.e. lies in C. This is a sheaf code in the *homological* (chain) direction, $\ker \partial_\ell$, analogous to the cohomological $\ker \delta^\ell$ of §5 but distinct from it: both are classical codes defined as kernels of a sheaf-boundary operator, but acting on chains and cochains respectively. The word "dual" would be misleading here — $\ker \partial_1^{\text{Tanner}}$ and $\ker \delta^\ell$ are not each other's dual codes in the coding-theoretic sense; the relationship is chain-vs-cochain, not code-vs-dual-code. Both are valid instances of the general framework; the chain direction is natural here because the local-code constraint is a *boundary* condition at each vertex. Stalk dimensions vary (1 on edges, $\Delta - \dim C$ on vertices in the quotient), and the restriction maps are nontrivial, so this is a non-constant, homogeneous sheaf on a non-Euclidean complex.

- **Left–right Cayley complexes / good classical LTCs** [13]. The *left-right Cayley complex* $X(G, A, B)$ of a group G with left-generator set A and right-generator set B has vertex set G, two families of directed edges ($g \xrightarrow{a} ag$ for $a \in A$ and $g \xrightarrow{b} gb$ for $b \in B$), and one square (g, a, b) for each $(g, a, b) \in G \times A \times B$, with corners g, ag, gb, agb. This is a 2-dimensional complex, not a product of two graphs; its squares arise from the commutation of left-A and right-B multiplication, not from a graph-product construction. Each square receives a local code that is a tensor product $C_A \otimes C_B$, and restrictions to incident edges are given by row and column projections. The sheaf structure encodes the Tanner-style local constraints that yield constant-rate, constant-distance, LDPC locally testable codes.

- **Lifted-product quantum LDPC codes** [11]. Starting from two classical LDPC codes with parity-check matrices H_A and H_B, the *lifted product* construction forms a CSS code whose X- and Z-check matrices are derived from a bilinear tensor-product complex over a cyclic group. The underlying complex is a 2-dimensional chain complex whose cells carry stalks drawn from H_A and H_B; the sheaf-code perspective is that these stalks and their restriction maps are precisely the data of a non-constant cellular sheaf, and the CSS code is the

kernel of the associated coboundary operators. Whether the logical space is naturally identified with H^1 of a specific sheaf on this complex is a question that requires constructing the relevant sheaf explicitly and verifying the coboundary maps; the PK paper establishes the code parameters directly via algebraic combinatorics. The construction achieves constant rate and linear distance.

- **Quantum Tanner codes** [12]. Same square complex; CSS with X-checks imposing the local code $C_A \otimes C_B$ at each square and Z-checks imposing $C_A^\perp \otimes C_B^\perp$. Constant rate and linear distance.
- **Higher cubical / sheaf-coefficient codes** [14]. t-dimensional cubical complex with an arbitrary cellular sheaf as coefficients; the qLDPC and qLTC theorems become coboundary-expansion statements for sheaves on X.

7.3 The unified picture

Strip the surface differences and every construction above is the same composite:

$$\underbrace{\text{non-Euclidean expander-type cell complex}}_{\text{geometry, kills BPT}} \times \underbrace{\text{nontrivial-algebra local stalks}}_{\text{algebra, gives the checks teeth}}$$

glued by the restriction maps of a cellular sheaf. Three things have to hold simultaneously, and the cellular-sheaf framework supplies all three in a single package:

- the cell complex must escape the BPT ceiling — this is the *geometry*;
- the local code structure at each cell must be nontrivial — tensor-product, Reed–Solomon, or BCH codes, encoded either as the stalk itself (Convention A) or as a local check on an ambient stalk (Convention B) — this is the *algebra*;
- the stalks at adjacent cells must be linked by restriction maps that satisfy $\delta^2 = 0$ on the cochain complex — this is the *glue*.

Drop the geometry: BPT bites and rate or distance vanishes. Drop the algebra: the checks enforce nothing and expansion gives nothing. Drop the glue: there is no chain complex, no CSS commutation, no notion of a logical operator. Each modern construction in the tour above instantiates all three simultaneously, and the cellular-sheaf formalism is the common language in which all three can be stated and composed. Whether it is *minimal* in a formal sense — i.e., whether every good qLDPC construction must involve sheaf-like structure — is a precise open question; what the tour shows is that every known construction fits naturally into this framework and that the two failure modes of §7 are genuinely obstructed by the absence of one or the other ingredient.

8 A tiny worked example beyond the toric code

To illustrate concretely how non-constant stalks and nontrivial restriction maps arise from a polynomial evaluation family (Convention A of §4.3), here is the smallest such example. The construction is structurally analogous to the U2 upgrade — the face stalk carries bilinear polynomial data and the restriction maps are evaluations — but lives on a single square rather than

a left-right Cayley complex, so it captures the local algebra of U2 without the group-theoretic global structure that gives U2 its expansion and distance. Take X to be a single *square* with four vertices, four edges, and one face, and set $q = 3$. Pick two distinct $\mathbb{F}_3$-coordinates α_0, α_1 for the horizontal axis and two distinct coordinates β_0, β_1 for the vertical axis; label the vertex at column i, row j as $v_{ij} = (\alpha_i, \beta_j)$, each horizontal edge by its row β_j, and each vertical edge by its column α_i. Define stalks

$$\mathcal{F}(v_{ij}) = \mathbb{F}_3, \qquad \mathcal{F}(\text{horiz edge at } \beta_j) = \mathbb{F}_3[s]_{\leq 1} \cong \mathbb{F}_3^2,$$
$$\mathcal{F}(\text{vert edge at } \alpha_i) = \mathbb{F}_3[t]_{\leq 1} \cong \mathbb{F}_3^2, \qquad \mathcal{F}(\text{face}) = \mathbb{F}_3[s,t]_{\leq (1,1)} \cong \mathbb{F}_3^4,$$

i.e. degree-≤ 1 forms in the axis variable on each edge, and a degree-$(1,1)$ bilinear form on the face. Restrictions are given by evaluation:

$$\rho_{\text{face}\to\text{horiz edge at } \beta_j} : f(s,t) \mapsto f(s,\beta_j), \qquad \rho_{\text{face}\to\text{vert edge at } \alpha_i} : f(s,t) \mapsto f(\alpha_i,t),$$
$$\rho_{\text{horiz edge}\to v_{ij}} : g(s) \mapsto g(\alpha_i), \qquad \rho_{\text{vert edge}\to v_{ij}} : h(t) \mapsto h(\beta_j).$$

Sheaf transitivity holds on the nose: each composite face $\to$ edge $\to v_{ij}$ and the direct face $\to v_{ij}$ both send $f(s,t)$ to $f(\alpha_i, \beta_j)$, so the cellular-sheaf axioms are satisfied. The chain identity $\partial_1 \partial_2 = 0$ for the sheaf boundary operator follows from two independent ingredients: first, the standard incidence arithmetic of the square ($\sum_\tau [\sigma : \tau][\tau : \nu] = 0$ for each face–vertex pair (σ, ν)), which holds for any oriented cell complex regardless of coefficients; second, the sheaf compatibility just verified, which allows the two restriction-map composites $\rho_{\text{face}\to e\to v_{ij}}$ and $\rho_{\text{face}\to v_{ij}}$ to be identified, ensuring the boundary-of-boundary lands in the correct stalk.

Code parameters. With $\alpha_0 = 0$, $\alpha_1 = 1$, $\beta_0 = 0$, $\beta_1 = 1$ in $\mathbb{F}_3$, the resulting cochain complex has

$$C^0 = \mathbb{F}_3^4, \quad C^1 = \mathbb{F}_3^8, \quad C^2 = \mathbb{F}_3^4,$$

with $n = 8$ (block length, stalkwise: four edges each contributing a 2-dimensional stalk). The coboundary $\delta^1 = \partial_2^T : C^1 \to C^2$ is a 4×8 matrix; one verifies that the four columns of ∂_2 (the restriction images of the four face-stalk basis vectors) are linearly independent over $\mathbb{F}_3$, so $\operatorname{rank}(\delta^1) = 4$. Hence

$$\dim \ker \delta^1 = 8 - 4 = 4, \qquad R = \tfrac{4}{8} = \tfrac{1}{2}.$$

For the cocycle distance: the four linear conditions imposed by $\delta^1 = \partial_2^T$ on $x = (x_1, \ldots, x_8) \in C^1 = \mathbb{F}_3^8$ (indexing the stalk components as x_1, x_2 for e_{h0}; x_3, x_4 for e_{h1}; x_5, x_6 for e_{v0}; x_7, x_8 for e_{v1}, with each pair encoding the constant and linear coefficient) are:

$$\begin{aligned}
&\text{(i)} && x_1 + 2x_3 + 2x_5 + x_7 = 0,\\
&\text{(ii)} && x_2 + 2x_4 + x_7 = 0,\\
&\text{(iii)} && 2x_3 + 2x_6 + x_8 = 0,\\
&\text{(iv)} && 2x_4 + x_8 = 0.
\end{aligned}$$

Checking each of the 8 coordinate basis vectors e_i (and their nonzero $\mathbb{F}_3$-multiples, which have the same support): e_1 fails (i) since $x_1 = 1 \neq 0$; e_2 fails (ii); e_3 fails (i) since $2x_3 = 2 \neq 0$; e_4 fails (ii) since $2x_4 = 2 \neq 0$; e_5 fails (i) since $2x_5 = 2 \neq 0$; e_6 fails (iii) since $2x_6 = 2 \neq 0$; e_7 fails (i) since $x_7 = 1 \neq 0$; e_8 fails (iii) since $x_8 = 1 \neq 0$. Hence no weight-1 element lies in $\ker \delta^1$. The vector $x = e_1 + e_5 = (1,0,0,0,1,0,0,0)$ (constant-coefficient slots of e_{h0} and e_{v0}, both equal to

1) satisfies all four conditions: (i) $1+2=3=0$; (ii) 0; (iii) 0; (iv) 0. This gives $d_{\mathrm{cyc}}=2$ and the classical $[8,4,2]_3$ code.

For the quantum code: the square X is contractible. We recall that for *any* cellular sheaf $\mathcal{F}$ of $\mathbb{F}$-vector spaces on a contractible finite CW complex, $H^k(X;\mathcal{F})=0$ for all $k>0$; this follows because the nerve of the face poset of X is homeomorphic to the barycentric subdivision of X (hence contractible), and the cohomology of any functor on a small category with contractible classifying space vanishes in positive degrees (see e.g. [9] for the sheaf- cohomology setup; the acyclicity statement is standard in poset cohomology). Applying this to $\mathcal{F}$ gives $H^1(X;\mathcal{F})=0$ — all 1-cocycles are exact, and the cohomological distance d_{coh} is undefined (there are no nontrivial logical operators). This can also be verified directly: with the explicit matrices above, $\operatorname{rank}(\delta^1)=4=\dim C^2$, so $H^2=0$; the Euler-characteristic identity $\dim C^0-\dim C^1+\dim C^2=4-8+4=0$ then forces $\dim H^0-\dim H^1=0$, and $\operatorname{rank}(\partial_1)=4$ gives $H^0=0$, hence $H^1=0$. The example therefore illustrates the non-constant sheaf structure of $\mathcal{F}$ and the resulting classical code, not a quantum code. Quantum codes with nonzero H^1 arise when the same sheaf structure is tiled over a non-simply-connected complex with good expansion: replacing the single square by the left-right Cayley complex $X(G,A,B)$ provides the spectral gap needed for linear d_{coh}. The nonvanishing of $H^1(X(G,A,B);\mathcal{F})$ for the relevant non-constant sheaf $\mathcal{F}$ is an *algebraic* fact, not a topological one. CSS commutation ($\delta^\ell\delta^{\ell-1}=0$) guarantees only that $\operatorname{im}\delta^{\ell-1}\subseteq\ker\delta^\ell$ (every coboundary is a cocycle), which is necessary but not sufficient for $H^\ell\neq 0$; strict inclusion $\operatorname{im}\delta^{\ell-1}\subsetneq\ker\delta^\ell$ requires a separate rank computation. For the left-right Cayley complex with codes C_A, C_B, that rank computation depends on the specific group G and the algebraic structure of C_A and C_B; it is carried out in [13, 12] and does not follow from topological or commutation considerations alone.

The *linear distance* $d_{\mathrm{coh}}=\Omega(n)$ requires two expansion conditions, one for each Pauli type:

- *X-type (coboundary expansion).* Every 1-cocycle of small support already lies in $\operatorname{im}\delta^0$ (is exact), so any nonzero class in $H^1=\ker\delta^1/\operatorname{im}\delta^0$ has a representative of large weight. This bounds the X-distance from below.
- *Z-type (boundary expansion).* Every 1-cycle of small support already lies in $\operatorname{im}\partial_2$ (is a boundary), so any nonzero class in $H_1=\ker\partial_1/\operatorname{im}\partial_2$ has a representative of large weight. This bounds the Z-distance from below.

These are logically independent properties: a complex can satisfy one without the other. Each requires its own expansion argument (established in [13, 12]), and neither follows from the other or from $H^1\neq 0$ alone. Both must be verified for the quantum code to achieve linear d_{coh} simultaneously in the X and Z directions.

Further reading. For the breakthrough qLDPC constructions, the canonical references are Panteleev–Kalachev [11], Leverrier–Zémor [12], Dinur–Evra–Livne–Lubotzky–Mozes [13], and Dinur–Lin–Vidick [14]. For sheaf-theoretic transversal non-Clifford gates, see Lin [15].

References

[1] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1977.

[2] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.

[3] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54(2):1098–1105, 1996.

[4] A. M. Steane. Multiple-particle interference and quantum error correction. *Proc. Roy. Soc. Lond. A*, 452:2551–2577, 1996.

[5] S. Bravyi and B. Terhal. A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes. *New Journal of Physics*, 11:043029, 2009.

[6] S. Bravyi, D. Poulin, and B. Terhal. Tradeoffs for reliable quantum information storage in 2D systems. *Phys. Rev. Lett.*, 104:050503, 2010.

[7] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27(5):533–547, 1981.

[8] M. Sipser and D. A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42(6):1710–1722, 1996.

[9] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

[10] R. Ghrist and C. Ding. Precision-graded cohomology and arithmetic persistence for network sheaves. Preprint, arXiv:2511.00677, 2025.

[11] P. Panteleev and G. Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, pages 375–388, 2022.

[12] A. Leverrier and G. Zémor. Quantum Tanner codes. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2022)*, pages 872–883, 2022.

[13] I. Dinur, S. Evra, R. Livne, A. Lubotzky, and S. Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, pages 357–374, 2022.

[14] I. Dinur, T.-C. Lin, and T. Vidick. Expansion of higher-dimensional cubical complexes with application to quantum locally testable codes. Preprint, arXiv:2402.07476, 2024.

[15] T.-C. Lin. Transversal non-Clifford gates for quantum LDPC codes on sheaves. Preprint, arXiv:2410.14631, 2024.